

화학공정의 Global Optimization을 위한 Interval Analysis Algorithm

한주란, 최수형, Vasilios Manousiouthakis*
진북대학교 화학공학과, Dept. of Chem. Eng., UCLA, U.S.A.*

An Interval Analysis Algorithm for Global Optimization of Chemical Processes

Ju Ran Han, Soo Hyung Choi, and Vasilios Manousiouthakis*
Dept. of Chem. Eng., Chonbuk National University
Dept. of Chem. Eng., UCLA, U.S.A.*

서론

대부분의 화학공정 최적화 문제는 Nonconvex NLP (Nonlinear Program) 형태로 나타나며, 많은 경우 여러개의 local optimum들이 존재한다. 그러나 현재 화학공정 최적화는 대부분 Local Optimization에 의존하고 있는 실정이다. 진정한 최적점을 찾는 Global Optimization의 방법은 크게 Deterministic Approach와 Stochastic Approach로 나눌 수 있다. 본 연구의 목적은 최적화 결과가 global optimum임을 이론적으로 확실하게 보장하는 Deterministic Approach에 의한 화학공정 최적화의 가능성을 규명하는 것이다.

Vaidyanathan과 El-Halwagi (1994)는 기존의 Interval Analysis Algorithm의 가속을 위하여 lower bound test, distrust-region method, shifted partitioning around local solutions의 방법들을 제안하였다. 본 논문에서는 등식 제약조건을 이용하여 보다 적은 branching variable들을 사용하는 algorithm을 제시한다. 이 algorithm은 마치 변수 치환을 통하여 문제의 크기를 줄인 것과 같은 가속효과를 낸다. 그러나 경우에 따라서 수렴하지 않을 수도 있다 (수렴하기 위한 충분조건은 아래 참조). 단, 수렴하여 결과가 얻어졌다면 그것은 global optimum이다.

이론

본 논문에서 제시하는 algorithm은 Ratschek과 Rokne의 Algorithm 2 (1988, pp. 174-176)에 기초를 둔다. 그들의 algorithm은 모든 변수를 branching 대상으로 고려한다. 그러나 많은 경우 일부 변수들의 범위는 다른 변수들의 범위로부터 결정될 수 있다. 이 경우 문제의 크기를 축소할 수 있으므로 이를 이용하면 Branch and Bound 과정의 계산 요구량을 현저히 줄일 수 있다. 따라서 제안되는 algorithm은 다음과 같은 형태의 최적화 문제를 다루기로 한다.

$$\min_{x, y} f(x, y)$$

subject to

$$g(x, y) \leq 0$$

$$h(x, y) = 0$$

$$x \in X = \{x \mid x \in R^{n_x}, -\infty < x_j^L \leq x_j \leq x_j^U < +\infty, j = 1, \dots, n_x\}$$

$$y \in Y = \{y \mid y \in R^{n_y}, -\infty < y_j^L \leq y_j \leq y_j^U < +\infty, j = 1, \dots, n_y\}$$

여기에서 $f, g,$ 및 h 는 연속 함수들이다. 변수 x 및 y 는 x 의 범위 X 로부터 위의 제약조건들을 통하여 y 의 범위 Y 가 결정될 수 있도록 선택한다. 단, 다음 조건이 만족되도록 한다.

Property (Y): For any $X^k \subset X$, as $w(X^k) = \text{ub } X^k - \text{lb } X^k \rightarrow 0, w(Y^k) \rightarrow 0.$

이 조건이 성립하면 Ratschek과 Rokne 의 수렴성 증명 (1988, pp. 181-183)이 본 algorithm에도 유효하다. 만약 이 조건을 만족하지 않는 y 변수를 사용하면 수렴성은 보장되지 않는다. 그러나 이 경우에도 수렴했다면 결과의 globality는 보장된다. 제안되는 algorithm은 다음과 같다.

Interval Analysis Branch and Bound Algorithm

Step 0. Initialization

- (1) Define $X^1 = X.$
- (2) Initialize the set of unsolved subproblem indices $B \leftarrow \{1\}$ and the set of solved subproblem indices $D \leftarrow \emptyset.$
- (3) Set the upper bound on the global minimum $U \leftarrow +\infty$ or $f(x^0, y^0),$ where (x^0, y^0) is any feasible point.

Step 1. Interval Analysis and Bounding. For all $k \in B,$

- (1) Remove k from $B,$ and enter k into $D.$
- (2) Determine Y^k corresponding to $X^k.$
- (3) Set $Y^k \leftarrow Y^k \cap Y.$
- (4) If $Y^k = \emptyset$ or $\text{lb } f(X^k, Y^k) > U$ or $\text{lb } g_i(X^k, Y^k) > \epsilon$ or $\text{lb } h_i(X^k, Y^k) > \epsilon$ or $\text{ub } h_i(X^k, Y^k) < -\epsilon$ for some $i,$ then remove k from $D.$
- (5) Otherwise,
 - (a) Set $x^k = \text{mid } X^k.$
 - (b) Determine y^k corresponding to $x^k.$
 - (c) If $y^k \in Y^k$ and $g_i(x^k, y^k) \leq \epsilon_f$ and $h_i(x^k, y^k) \in [-\epsilon_f, +\epsilon_f]$ for all $i,$ then
 - (i) Update $U \leftarrow \min(U, f(x^k, y^k)).$
 - (ii) Remove all r from D such that $\text{lb } f(X^r, Y^r) > U.$

Step 2. Convergence Test and Branching

- (1) If $D = \emptyset,$ terminate as the problem is infeasible.
- (2) Otherwise,
 - (a) Select $k \in D$ such that $\text{lb } f(X^k, Y^k) = \min_{r \in D} \text{lb } f(X^r, Y^r).$
 - (b) If $y^k \in Y^k$ and $f(x^k, y^k) - \text{lb } f(X^k, Y^k) \leq \epsilon_0$ and $g_i(x^k, y^k) \leq \epsilon_f$ and $h_i(x^k, y^k) \in [-\epsilon_f, +\epsilon_f]$ for all $i,$ then terminate with (x^k, y^k) as a global solution.
 - (c) Otherwise,
 - (i) Select variable x_j which has the maximum length of interval in $X^k.$
 - (ii) Bisect X^k normal to coordinate $x_j,$ getting X^p and X^q such that $X^k = X^p \cup X^q,$ where $p, q \notin D.$
 - (iii) Remove k from $D,$ and enter p and q into $B.$
 - (iv) Go to Step 1.

위의 algorithm 중 Step 1(4)에서 infeasibility test를 위하여 사용되는 ϵ 은 0 이상의 작은 값으로, lb (lower bound) 및 ub (upper bound) 계산시 machine interval arithmetic (Ratschek and Rokne, 1988, pp. 30-31)을 사용할 경우 0으로 놓을 수 있다. 본 연구에서는 $\epsilon = 10^{-10}$ 이 사용되었다.

본 연구에서 개발된 최적화 시스템은 다음과 같은 형태의 y 변수 갯수만큼의 등식 제약조건을 사용하여 변수 x 의 범위로부터 변수 y 의 범위를 구한다.

$$A(x)y = b(x) \quad (1)$$

여기에서 A 는 $n_y \times n_y$ matrix, b 는 n_y -dimensional column vector이며, 이들의 각 원소는 상수 또는 x 의 함수들이다. 만일 어떤 주어진 x 의 범위에 대하여 행렬 A 의 범위가 singular matrix를 포함하면 Property (Y)는 성립하지 않는다. 이 경우에도 본 algorithm은 수렴할 수 있으며 결과는 유효하다. 만일 수렴하지 않는다면, y 변수의 갯수를 줄이거나, 식 (1)의 symbolic solution을 이용하는 변수 치환을 통한 문제 변형을 시도할 수 있다.

예 제

다음은 Stephanopoulos와 Westerberg (1975)의 문제이다.

$$\min x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6$$

subject to

$$\begin{aligned} -3x_1 + x_2 - 3x_4 &= 0 \\ -2x_2 + x_3 - 2x_5 &= 0 \\ 4x_4 - x_6 &= 0 \\ x_1 + 2x_4 &\leq 4 \\ x_2 + x_5 &\leq 4 \\ x_3 + x_6 &\leq 6 \\ x_1 &\leq 3 \\ x_3 &= 4 \\ x_5 &\leq 2 \\ x_1, x_2, x_3, x_4, x_5, x_6 &\geq 0 \end{aligned}$$

Vaidyanathan과 El-Halwagi (1994)는 다음과 같은 조건을 사용하였다.

$$\text{Initial box} = ([0, 3], [0, 3], [0, 4], [0, 3], [0, 3], [0, 12])^T.$$

$$\text{Tolerance on the width of the solution box} = 0.001$$

$$\text{Accuracy of the objective function inclusion} = 0.01$$

최적점의 범위는 $([0.166, 0.167], [1.999, 2.000], [3.999, 4.000], [0.499, 0.500], [0.000, 0.000], [1.999, 2.000])^T$, 목적함수의 범위는 $[-13.402, -13.393]$ 이었다. 계산 시간은 그들의 모든 가속 방법들을 도입하였을 때 Sun SPARCstation 10 컴퓨터에서 436.4 s 인 것으로 보고되었다.

본 연구에서도 동일한 initial box를 사용하였으며, 보다 정확한 해를 얻기 위하여 global minimum에서의 제약조건을 허용오차 (feasibility tolerance) $\epsilon_f = 10^{-4}$ 및 목적함수의 허용오차 (optimality tolerance) $\epsilon_o = 10^{-4}$ 를 사용하였다.

최적점은 위와 동일하며 목적함수의 범위는 $[-13.402, -13.401]$ 로 나타났다. 계산 시간은 제안된 가속 방법을 사용하지 않았을 때 PC 486DX2/66MHz에서 4.95 s였다. 변수 y 를 사용함으로써 얻어지는 가속 효과는 표 1에 요약되어 있다.

표 1. 예제에서 y 변수 사용에 따른 계산 시간 변화.

x 변수	y 변수	Subproblem 갯수	계산 시간, s
$x_1, x_2, x_3, x_4, x_5, x_6$		17,267	4.95
x_1, x_2, x_3, x_4, x_5	x_6	5,263	1.60
x_1, x_2, x_3, x_4	x_5, x_6	2,655	0.88
x_1, x_2, x_3	x_4, x_5, x_6	925	0.44
x_1, x_2	x_3, x_4, x_5, x_6	275	0.28

결론

일반적으로 Nonconvex NLP는 NP-hard problem이다. 즉, 문제의 크기가 증가하면 계산 시간은 대개 지수 함수 형태로 증가한다. 화학공정 최적화 문제는 대부분 nonconvex한데다 그 크기가 방대하여 기존의 Deterministic Approach를 사용하는 Global Optimization Algorithm은 실효성이 거의 없었다.

그러나 대부분의 화학공정 최적화 문제들은 많은 등식 제약조건을 가지며, 문제에 따라서는 많은 등식을 식 (1)의 형태로 나타낼 수 있다. 본 논문의 예제와 같은 최선의 경우, branching variable의 갯수를 문제의 자유도와 같게 할 수 있었다. 이러한 특성을 가진 문제는 사실상 작은 문제라 할 수 있다.

대부분의 화학공정 최적화 문제의 자유도는 그리 크지 않다. 따라서 본 논문에서 제안한 algorithm을 그동안 Local Optimization이나 Stochastic Approach에만 의존하던 많은 문제들에 적용시킴으로써 이론적으로 보장된 진정한 최적점을 찾거나 확인할 수 있을 것으로 판단된다.

감사의 글

본 연구를 지원한 미국 Ralph M. Parsons Foundation (grant no. L900926) 및 공정산업의 지능자동화연구센터(한국과학재단 ERC)에 감사드립니다.

참고문헌

- Ratschek, H., and J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood Limited, England (1988).
- Stephanopoulos, G., and A. W. Westerberg, "The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization," *JOTA* 15, 285 (1975).
- Vaidyanathan, R., and M. El-Haiwagi, "Global Optimization of Nonconvex Nonlinear Programs via Interval Analysis," *Comput. Chem. Eng.*, 18(10) 889-897 (1994).